

# Integration of UAVs with Real Time Operating Systems using UAVCAN

Niranjan Ravi

*Electrical and Computer Engineering  
Purdue School of Engineering and Technology  
Indianapolis, USA  
ravin@iu.edu*

Mohamed El-Sharkawy

*Electrical and Computer Engineering  
Purdue School of Engineering and Technology  
Indianapolis, USA  
melshark@iupui.edu*

**Abstract**—In today's world, the applications of Unmanned Aerial Vehicle (UAV) systems are leaping by extending their scope from military applications on to commercial and medical sectors as well. Owing to this commercialization, the necessity to append external hardware with UAV systems becomes inevitable. This external hardware could aid in enabling wireless data transfer between the UAV system and remote Wireless Sensor Networks (WSN) using low powered architecture like Thread, BLE (Bluetooth Low Energy). The data is being transmitted from the flight controller to the ground control station using MAVLink (Micro Air Vehicle Link) protocol. The ideal aim of this research is to address the issues of integrating different hardware with the flight controller of the UAV system using a light-weight protocol called UAVCAN. This approach would result in reduced wiring and would harness the problem of integrating multiple systems to UAV.

**Index Terms**—IEEE 802.15.4, 6LoWPAN, BLE, Thread Protocol, UAV, MAVLink

## I. INTRODUCTION

The Internet has become a part of every day's life to gather and process the information collected across the globe. The development in various fields of technology fostered rapid growth in the area of research and paved a new path for homogeneous as well as heterogeneous modes of communication between humans and things[1]. This approach helped researchers to build new solutions for problems faced by human society. The advent of WSN, made the applications of embedded systems have been more feasible. A WSN is a small network of spatially distributed autonomous nodes that communicate with others in the network using radio signals. WSN consists of many independent low cost, micro-sensor nodes capable of sensing, processing the data, and communicating them further. The micro-sensor node would consist of a sensor, a microcontroller or a microprocessor, a wireless transmitter with an inbuilt power source[2][3]. One of the ideal objectives of a WSN is sensing the environmental data like temperature, air pollution level in their respective locations. When employing WSN for data collection, a variety of approaches such as BLE, Thread, and ZigBee are used to transmit the data wirelessly between the WSN nodes without

the use of the internet. This approach would help to develop more low powered architecture since it would work even if there is no internet. Wireless Mesh Network like Thread protocol is very efficient when information has to be transmitted from remote WSN to other locations because it provides no single point of failure, which ensures no data loss during the process[4]. Also, WSNs are characterized by diversity because there are many different proprietary and non-proprietary solutions. At the same time, the advent of UAV systems has drastically helped humankind in various ways. These remotely piloted aircraft are employed in various types of applications from being surgical strike drones in the military sector to commercial deployments like agriculture, public safety, and data collection from various terrains.

There has been a pressing need from industrial and medical sectors to make this WSN data readily available for advanced research activities in the areas where it is onerous to deploy human resources for data collection. For example, during the times of natural calamities, it is difficult to use human resources to the affected area. The real challenge arises when the need to transmit WSN data from remote locations to research destinations without the usage of the internet, cause the signal towers may get affected during crisis.

The ideal aim of this research is to propose a system with the help of UAV and Real-Time Operating System (RTOS) to address these challenges as mentioned earlier. UAV systems are growing at a rapid rate in various aspects of life, including the dispatch of medicines and undergo video surveillance during an emergency due to less air traffic. The ability of UAV to follow a pre-programmed flight path also aids in the design since they give the ability for UAV to travel without any barriers and irrespective of the terrains. UAVs are used to collect the data which has been gathered by WSN at remote locations where internet connectivity is not predominant. The architecture consists of a UAV system appended with an RTOS using UAVCAN (Unmanned Aerial Vehicle Controller Area Network). This enables the UAV system to use Thread Protocol enabled in RTOS to receive the data wirelessly without the internet.

## II. RELATED STUDY ON REAL TIME OPERATING SYSTEMS

The field of embedded systems has been growing at an Exponential rate over the last few years. The number of embedded systems employed in our day to day life is increasing at a rapidly ranging from a coffee maker to autonomous cars development of small, compact embedded systems capable of performing various operations in real-Time. This is called multi-tasking. But owing to the processing of multiple processes at the same time, a need for an operating system to work in real time had developed this Paved way to the development of RTOS. RTOS is defined to be a software component that rapidly between the tasks, which gives the impression that multiple programs are executed simultaneously on a single processing core. But on the contrary, the processor can run a single thread at a time, but RTOS decides the priority of the task, which aids the processor to switch between various tasks[5].

### A. Interrupt Latency

Real-time systems are designed to carry out tasks/thread within an absolute worst-case scenario in terms of time Complexity. This would result in creating a system with a higher degree of reliability when compared to an operating system[6].

### B. Resource constrained processors

Microkernels in RTOS systems deliver a hard real-time response which places a crucial role in embedded microprocessors with limited RAM/ROM[5].

### C. Priority Scheduling

Real-time operating systems schedule the tasks based on the priority level decided by designers. High priority tasks are given 100 percent importance compared to a low priority task[6].

### D. Improved Efficiency

RTOS is an event-driven system. These systems would increase efficiency by reducing the processing time on the tasks which never occurred during the run-time of the processor[6].

## III. BACKGROUND STUDY ON INTERNET OF THINGS

### A. Advancements in field of embedded systems

The field of embedded systems is advancing at a constant pace in day to days life. While designing an embedded system could be relatively simple, but the features and results provided by embedded systems are beyond what a human could offer. For example, aviation systems are constructed as an embedded system to integrate sensor data and provide real-time sensor output. This approach created a path to explore the data from embedded systems by a new feature called the Internet of things (IoT).



Fig. 1. Internet of Things[14]

### B. IoT

The term IoT was coined initially by Auto-ID center in 1999, which visualized the future where every physical the object would be tagged by installing a radio frequency identification (RFID) tag which would have a unique identifier globally[1]. Internet of things is a platform for a the network of embedded physical devices are built by electronic components, software, sensors and establishing a two-way communication between the connected devices in a system or with the manufacturer. The term IoT commonly refers to a collection of devices that are generally resource-constrained in terms of computation and communication capabilities. This structure has been dramatically modified by establishing a connection between all IoT devices to the internet and various services, and features. This would result in the development of more machine to machine communication using the internet without any human intervention. IoT also paves the way for connecting many heterogeneous networks in which each system adopts a different communication pattern such as human-to-human (H2H), thing-to things (t2T) or thing-to-thing (t2T). So, this would give an advantage of monitoring the devices in case of hardware and software issues remotely and also aids in analyzing the data over the internet. The concept of IoT has widely expanded, and now it encloses a wide variety of technologies and communication protocols. The primary reason for the interest of the research community to redesign and deploy new enhancements into the field of IoT. The embedded systems which are a part of IoT systems can understand and act according to the environment where they are located. They are referred to as smart objects or smart devices. The introduction of IPv6 and CoAP as basic blocks for IoT applications allows the establishment of the connection between IoT hosts to the internet[1]. That would amplify the advantages to the existing systems, including:

- Different systems can be integrated and work towards a

common goal.

- The development of various devices that vary in computational capacities.
- An integrated interface for applications that remove the needs for application-level procedures.

The above advantages simplify the development and deployment of smart devices in various real-time scenarios from automation projects to a real-time environment.

#### IV. RELATED STUDY ON UAV SYSTEMS

The rapid technological developments in the field of electronics and sensor technologies gave rise to the usage of UAVs in the field of both civilian and military operations[7]. UAV systems are deployed in an area where it is challenging to deploy military personnel. They can be used for surveillance of a city or could also use as a mode of air-strike on enemy targets. Scientists all over the world work on the usage of UAV systems in various aspects of life. This range from deploying a drone during a time of disaster to look for human survivors or even to have secret surveillance on enemy camp during the time of war. The primary reason for the uprising scalability of UAV systems is that they are easy to construct, economical, and their ability to do predetermined work. The principal activities of drones include:

- Environmental and scientific research.
- As a long-range weapon in military air strikes.
- Aerial surveillance.
- Public safety.

A typical UAV system would consist of a GPS module to help in the accurate navigation of the drone to the destined path and a telemetry kit which allows observing the flight data from the ground/base station. To obtain the flight dimensions in real time, the flight controllers embedded with an accelerometer, gyro meter, and magnetometer[16]. They also include a ground station such as a PC/laptop which would send and receive data from the drones through telemetry device. The ground control station gives an option to upload a predefined path for the UAV system. Another option is operating the drone in manual mode for testing and trial purpose[17].

#### V. PIXHAWK/PX4

Pixhawk is an open-source project developed initially by Swiss Federal Institute of Technology at Zurich for Micro Air Vehicle transportation. Pixhawk based on ARM Cortex M4 architecture with a clock frequency of 168 MHz, and it can be programmed through Universal Serial Bus (USB)[8].

##### A. PX4 Firmware

PX4 works on a lightweight and an energy efficient real-time operating system called NuttX, which provides a Portable Operating System Interface (POSIX) style environment. NuttX is designed for resource-constrained systems[9]. It provides a command line interface called NuttShell (nsh) which can be accessed by ground Control station to access the system. PX4 middleware also provides a



Fig. 2. Pixhawk Autopilot Board[8]

micro Object Request Broker (uORB) for an asynchronous mode of communication between individual tasks. uORB can help in data transfer between threads by a simple publish-subscription pattern[10].

#### VI. CAN COMMUNICATION STANDARD

Robert Bosch GmbH, German engineering, and electronics company initially established Controller Area Network (CAN) Bus standard in the automotive industry[11], to replace complex wiring harness with two wire bus. The CAN bus has immunity to electrical noise and repair data error. This standard gained popularity in various industries ranging from automobile to medical, industrial sectors[12]. CAN bus is a multi-master, message broadcast system that has a maximum signaling rate of about 1 Mbps. The messages or CAN frames broadcasted to the entire network, which provides data consistency in every node of the system. There are two types of the CAN standard:

- Standard CAN.
- Extended CAN.

CAN communication protocol is a Carrier-Sense Multiple Access protocol (CSMA) with Collision Detection and Arbitration on Message Priority (CD+AMP). It indicates every node in the network should wait for a particular period of time before transmitting a message. The collisions are resolved by a bit-wise arbitration, based on the priority of individual information/message in the identifier field of the CAN frame. The standard CAN have an 11-bit identifier with a signaling rate of about 125Kbps to 1 Mbps[16]. Later, was introduced extended CAN frame with a 29-bit identifier. The standard CAN provides 2048 unique message identifiers while the extended CAN provide 537 million identifiers. They both are compatible with each other and can work on the same bus.

##### A. CAN bus

In a system, logic-high associated with one and logic low associated with zero. But on the CAN bus, it works oppositely. The highest priority is given to the lowest bit ID. In the bus, the CAN-High and CAN-Low wires should be terminated by the resistance of about 120 ohms.

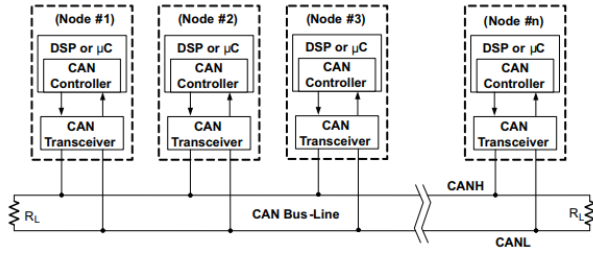


Fig. 3. CAN bus Network Configuration[11]

## B. UAVCAN

UAVCAN protocol is designed for reliable communication for aerospace and robotic applications via CAN bus[13]. UAVCAN acts as decentralized network like similar to a CAN bus where as each node would have a unique ID called Node identifier. Nodes aids in the process of data transfer using two methods:

- Message transfer - A message broadcast that contains a serialized message.
- Service transfer - A unicast transfer that contains either a service request or a service response.

Both modes of communication are defined using the Data Structure Description Language (DSDL). Message broadcasting is adopted in this research. DSDL is used to specify the data structures which have to be transmitted as a CAN frame. Each DSDL definition file would have a unique identifier followed by the data type name. Vendor-specific data types are also initialized in the system. A new data type name has to be added for each customer specific applications such that it does not affect the existing system functionality. In message broadcasting, messages are broadcasted can be distinguished into two types:

- Single Frame transfer - The payload is entirely transmitted in a single CAN frame up-to 8 bytes.
- Multi Frame transfer -The payload is divided and sent using CAN different frame.

Node Ids can be anonymous during the process of initialization. But anonymous node ID support single frame transfer method alone. Single frame transfers are more efficient than multi frame transfers in terms of latency and throughput. UAVCAN GUI tool is useful in understanding and visualizing the CAN frames in real-time.



Fig. 4. Single CAN frame Transfer[13]

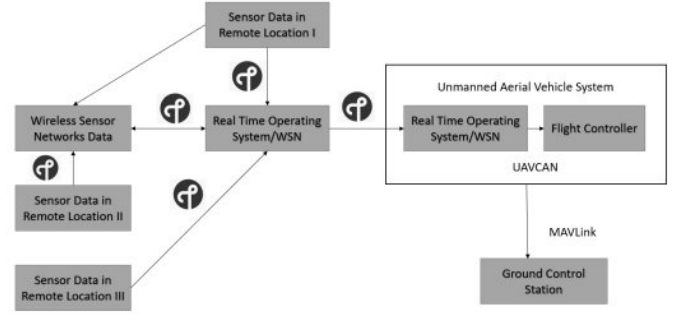


Fig. 5. Proposed Block Diagram For The System

## VII. PROPOSED MODEL OF INTEGRATING RTOS SYSTEM WITH UAV

There are various modes of wireless communication and data transfer in IoT devices these days. BLE and Thread network is used as a predominant wireless method of data transfer owing to the low powered architecture and no single point of failure and are cheap and can transfer data wirelessly up to 300 m without any internet. It would provide a crucial advantage during the times of crisis or disaster because the possibility of getting internet is very narrow. Having a clear understanding of RTOS system from the above topic comes handy because nowadays the majority of RTOS is enabled with BLE/Thread protocols. The principal aim of this research is to integrate an RTOS system with flight controller on the drone using UAVCAN, a light protocol designed for reliable communication in aerospace and robotic applications over robust vehicular networks such as CAN bus. The background study on the research article[15] aided in better understanding of UAVCAN principles for this research. This integration would enable the drone to acquire data wirelessly from various systems such as remote WSN locations which do not have internet or even from a different drone when it is in the air. Further, this data can be wirelessly sent to the base station using telemetry, a radio transmission.

## VIII. HARDWARE AND SOFTWARE REQUIREMENTS

- UAV system as Holybro Pixhawk- Flight controller.
- NXP's Rapid IoT - FreeRTOS Module.
- Zubax Babel -UAVCAN GUI Tool.
- External CAN Transreceivers.
- Hexiware Docking Station.
- Eclipse IDE.

## IX. SYSTEM ARCHITECTURE

The system design consists of a flight controller connected to Zubax babel to monitor UAVCAN/CAN frames using GUI tool and a rapid IoT attached to hexiwear docking station fitted with external CAN transreceivers. The flight controller and FreeRTOS system board are connected using CAN High and



CAN Low Pins. CAN High and CAN Low pins from Zubax babel is also attached to the above system.

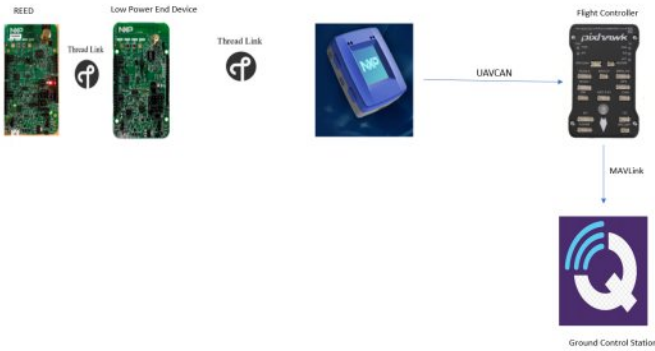


Fig. 6. System Architecture

## X. INITIAL SETUP AND CONFIGURATION

Pixhawk, an autopilot module is updated with latest version PX4 firmware from git repository. Two message files and two c files were created inside px4 firmware to test uORB communication using simple publish-subscribe method. One of the program files was used to send incremental integers every second and other file was used to receive the integers and store them in a variable. This behaviour is witnessed through Nuttx shell. Two new UAVCAN messages were created in px4 firmware. One was used to send data as a CAN frame to the external hardware and another messages was used to receive the incoming the CAN frames. Necessary changes were made in controller files such that it would not affect the existing UAVCAN functionality. To visualize the CAN frames in the flight controller, a UAVCAN GUI tool was used. The bus monitor feature was helpful to witness and decode the CAN frames in the device. Ardupilot software stack can also be used to debug errors in the CAN frames but GUI tool was adopted in this research for testing frames. To extend the range of data transmission, NXP's Rapid IoT, a freeRTOS system is used as a Thread Radio Module in this research. NXP's FRDM KW41Z is assumed to be the Thread Radio Module on the remote locations. Both these devices are ensured that they both exist on the same network. Rapid IoT being the leader and other being a remote end device which would transmit sensor data.

## XI. IMPLEMENTATION

The setup of the UAV system is carried out by addition of GPS for navigation of the drone, telemetry to send and receive information with ground control station by radio transmission. GPS and telemetry are connected to the respective connection ports of flight controller. The rapid IoT kit is programmed with the required network credentials mentioned in the initial setup and it is mounted on the UAV system with the help of hexiwear docking station. This

docking station provides ports for external interfaces to rapid IoT module. It is also fitted with external CAN transceiver to convert CAN TX and CAN RX data into CAN High and CAN Low signals. The CAN termination resistor is already existing in the transceiver. The UAV system is pre-programmed with an autonomous flight path to follow programmed travel directions. On the other hand, the KW41Z board located on remote location is programmed with the same set of network credentials that include unique network ID and network name. The remote thread module would receive the sensor data from remote wireless sensor networks. Once the UAV system reaches the remote locations such as a hilly area or a disaster location, this thread network in UAV system gets activated would request data from remote thread module within its range. The data packets are transferred wirelessly using UDP. The packets are transferred even from thread nodes which are not present in the immediate range of UAV system. Once the freeRTOS receives the data, it uses UAVCAN communication bus to transmit the data to flight controller which in turns sends the data through telemetry to ground station which can be about 2 miles away. The data can be seen in the ground Control shell terminal as well.



Fig. 7. UAV System with Rapid IoT

## XII. RESULTS AND DEMONSTRATION

The results are observed for two instances. During the integration of RTOS system, we can visualize the outputs in the terminal when the autopilot board sends CAN frames, and when the RTOS system transmits the data to autopilot board, it is observed in GUI tool.

## XIII. CONCLUSION

Technological growth and developments help mankind in different ways. It makes world a better place to live-in for all of us. But internet plays a major role in all these developments. In developed countries, having internet connectivity is a part of every-day's life and people depend on it almost for all activities in a day like using GPS for navigation to reach office using less traffic path and using internet even to order their favourite food. But it is not the same in developing or underdeveloped countries. There are

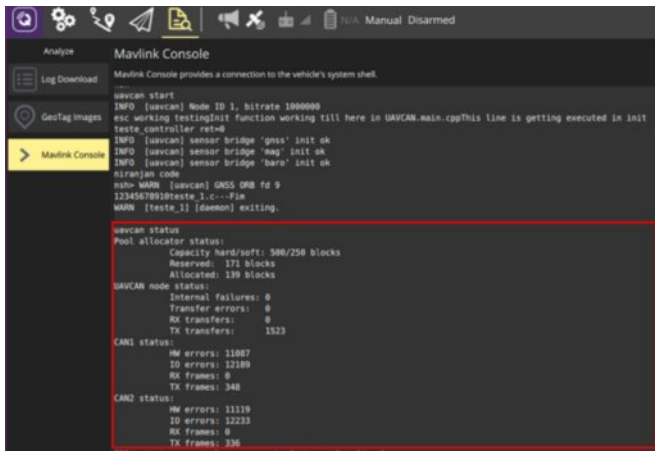


Fig. 8. UAVCAN Frame Status Through Ground Control

Dir	Local Time	CAN ID	Data Hex	Data ASCII	Src	Dst	TX/RX/FPS
RX	22:49:31.584352	1002F801	34 CB	4.	1	<unknown message 760>	12/1
RX	22:49:31.587353	1002F801	34 CC	4.	1	<unknown message 760>	
RX	22:49:31.588353	1002F801	34 CD	4.	1	<unknown message 760>	
RX	22:49:31.588353	1002F801	34 CE	4.	1	<unknown message 760>	
RX	22:49:31.588353	1002F801	34 CF	4.	1	<unknown message 760>	
RX	22:49:31.588353	1002F801	34 D0	4.	1	<unknown message 760>	
RX	22:49:31.589353	1002F801	34 D1	4.	1	<unknown message 760>	
RX	22:49:31.591354	1002F801	34 D2	4.	1	<unknown message 760>	
RX	22:49:31.592354	1002F801	34 D3	4.	1	<unknown message 760>	
RX	22:49:31.595354	1002F801	34 D4	4.	1	<unknown message 760>	
RX	22:49:31.596354	1002F801	34 D5	4.	1	<unknown message 760>	
RX	22:49:31.596354	1002F801	34 D6	4.	1	<unknown message 760>	
RX	22:49:31.596354	1002F801	34 D7	4.	1	<unknown message 760>	
RX	22:49:31.596354	1002F801	34 D8	4.	1	<unknown message 760>	
RX	22:49:31.597355	1002F801	34 D9	4.	1	<unknown message 760>	
RX	22:49:31.600356	1002F801	34 DA	4.	1	<unknown message 760>	
RX	22:49:31.600356	1002F801	34 DB	4.	1	<unknown message 760>	

Fig. 9. UAVCAN GUI Tool to Monitor CAN Frames

places in the world, where getting a network signal in the place they live happens once in a while. Such places, not all technological advancements are helpful. The ideal aim of this research is to use UAV systems to aid in development and provide immediate help to places where internet/network coverage is absent. To make this idea come true, wireless sensor networks are deployed in areas of interest. The regions can be remote locations where internet connectivity is not well established. It becomes difficult to establish connectivity in such regions. But UAVs are capable of reaching the remote locations where it is difficult for humans to reach. By incorporating thread protocol, a low power architecture to transfer data wirelessly several advantages are brought into UAV system. This system would pave a new way of data transportation which comes handy during natural crisis/disasters.

An ideal advantage of this research is establishing both the applications using UAVCAN protocol. The CAN bus reduces the number of wiring. The future of UAV system is projected to have CAN bus for all sorts of communication with external hardware since it is less susceptible to external noise. By integrating the flight controller with freeRTOS system using CAN bus, this research would provide background knowledge on UAVs in establishing two way data exchange between connected devices. This approach becomes efficient when many sensors have to be integrated

to the system design.

#### XIV. ACKNOWLEDGEMENT

We thank ECE department, Purdue School of Engineering and Technology, IUPUI for the hardware support required for the research and our colleagues from IoT Collaboratory lab. We also, like to show our gratitude to NXP and UAVCAN forum for sharing their knowledge and feedback for the research.

#### REFERENCES

- [1] S. Kumar, and M. Sethi. "State-of-the-Art and Challenges for the Internet of Things Security", (Last Accessed: April 24, 2019). [Online]. <https://tools.ietf.org/html/draft-irtf-t2trg-iot-seccons-05> (2018).
- [2] L. Mainetti, L. Patrono, and A. Vilei, Evolution of wireless sensor networks towards the internet of things: A survey, in SoftCOM 2011, 19th International Conference on Software, Telecommunications and Computer Networks, September 2011, pp. 16.
- [3] C. Rotariu, R. G. Bozomitu, V. Cehan, A. Pasarica, and H. Costin, A wireless sensor network for remote monitoring of bioimpedance, in 2015 38th International Spring Seminar on Electronics Technology (ISSE), May 2015, pp. 487490.
- [4] R.Niranjan and M. El-Sharkawy. "Collision avoidance and Drone surveillance using Thread protocol in V2V and V2I communications", pending publication.
- [5] RTOS, "What is an RTOS", (Last Accessed: March 23, 2019). [Online] Available: <https://www.highintegritysystems.com/rtos/what-is-an-rtos/>
- [6] RTOS, "Introduction to RTOS", (Last Accessed: February 14, 2019). [Online] Available: <https://www.ni.com/en-us/innovations/white-papers/07/what-is-a-real-time-operating-system-rtos-.html>
- [7] M. Farhan, "Efficient data communication in unmanned aerial vehicles", Thesis.35, United Arab Emirates University, 2015.
- [8] L. Meier et al., "PIXHAWK: A system for autonomous flight using onboard computer vision", Robotics and Automation (ICRA), 2011 IEEE International Conference on, 2011, pp. 2992-2997. International Conference on, 2011, pp. 2992-2997.
- [9] Dronecode, "PX4 overview", (Last Accessed: April 29, 2019). [Online] Available: <https://dev.px4.io/en/concept/architecture.html>
- [10] uORB, "uORB Messaging ", (Last Accessed: April 29, 2019). [Online] Available: <https://dev.px4.io/en/middleware/uorb.html>
- [11] CAN, "Controller Area Network", (Last Accessed: April 29, 2019). [Online] Available: <http://www.ti.com/lit/an/sloa101b/sloa101b.pdf>
- [12] CAN, "A simple Intro", (Last Accessed: April 29, 2019). [Online] Available: <https://www.csselecronics.com/screen/page/simple-intro-to-can-bus>
- [13] UAVCAN, "Complete background on UAVCAN", (Last Accessed: April 29, 2019). [Online] Available: <https://uavcan.org/>
- [14] IoT, "Overview Of Internet of Things", (Last Accessed: February 21, 2019). [Online] Available: <https://www.alibabacloud.com/>
- [15] UAVCAN, "Development of technology and procedures for health monitoring of UAV subsystems", (Last Accessed: June 5, 2019). [Online] Available: <https://fenix.tecnico.ulisboa.pt/downloadFile/563345090414487/Tese>
- [16] R. Niranjan and M. El-Sharkawy, Applications of Drones using Wireless Sensor Networks, pending publication.
- [17] V. Sivateja and M. El-Sharkawy, Remote Wireless Sensor Network Range Extension using UAVs with Thread Protocol (2018), pending publication.
- [18] W. Stallings. IPv6: the new Internet protocol, IEEE Communications Magazine, 34(7), pp. 96-108.
- [19] Hackaday, Choosing a flight controller, (Last Accessed: April 29, 2019). [Online] Available: <https://hackaday.com/2014/06/06/droning-on-ightcontroller-round-up/>
- [20] Pixhawk, What is pixhawk (Last Accessed: April 29, 2019). [Online] Available: <http://pixhawk.org/>